Supporting Conceptual Modeling
Bridging the Gap between Learner and System

Romi Geleijn
11044012

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligente

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor

dr B. Bredeweg

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

June 29th, 2018
Abstract

This research focuses on how effective help can be provided in Interactive Learning Environments for modeling. This help should increase learners' understanding of the model formalisms and subsequently of the subject matter. To this end, DynaLearn is expanded with explanations which shed light on how its underlying reasoning engine functions. Firstly, it is investigated how learners process the model cognitively. Additionally, the different types of explanation that benefit DynaLearn are identified based on instructional design research. Modeling errors that are common in DynaLearn are also considered. Based on the insights that follow from these sections, the explanations are designed. The four strategies ('what', 'why', 'why not' and 'how'-explanations) used to battle learners' misunderstanding of the reasoning engine are examined in terms of design choices and their implementation. The explanations are evaluated in three-fold; by testing the scope of the explanations, their technological effectiveness and their usability. The explanations help correct all identified modeling errors and offer a full coverage of all model configurations. Students are satisfied with the working of the display of the explanations, but request the content of the explanations to be extended to also cover instructions on how to navigate DynaLearn’s interface and how to fix modeling inconsistencies. In future research, DynaLearn’s explanations should be extended to further meet these students' wishes.
1 Introduction

In modern education the learner is considered to be an active sense maker, who organizes relevant information into a coherent structure and integrates it with pre-existing knowledge (Clark & Mayer, 2016). This philosophy is emphasized by the introduction of Interactive Learning Environments (ILE), which supply learners with hands-on experience with the learning material and promote learning by doing. ILE’s are defined as “computer-based instructional systems that offer a task environment and provide support to help learners develop skills or understand concepts involved in that task” (Aleven et al., 2003).

1.1 DynaLearn

DynaLearn is an ILE that enables learners to develop a better understanding of the subject matter by creating conceptual representations of complex, dynamic systems (Bredeweg et al., 2013). DynaLearn can be used in a range of subjects that require modeling (physics, chemistry, biology, economics, etc.). The modeling environment contributes to the curriculum by allowing learners to express knowledge in the form of a model and confronting them with the logical consequences of this model through simulating the system’s behaviour. It offers a graphical interface where learners can manipulate icons and express relationships using a diagrammatic representation (figure 1). In these simulations time is represented as a graph of states, where state transitions occur when values of quantities change. A state having multiple transitions signifies ambiguity.

![Figure 1: An example of a simple model in DynaLearn in the standard modeling level with labeled elements. It shows a habitat inhabited by predators (Beer) and prey (Vis). The population size of the predator is increasing, which causes the population size of the prey to decrease, as there is a negative relation between the population size of the predator and the prey. To the right the learner can view the different states resulting from the simulation.](image_url)
The logical consequences of the model are determined by deploying the theory of Qualitative Reasoning (Bredeweg et al., 2013). Qualitative Reasoning is a research field within Artificial Intelligence that represents conceptual notions in a way that closely matches human reasoning (De Koning et al., 2000), while also being grounded in mathematical formalisms allowing for automated computation. Instead of incorporating numerical information, qualitative models represent a system by including cause-effect relationships and qualitative information that expresses the magnitude (for example ‘high’ and ‘low’) and the direction of change.

1.2 Previous DynaLearn research

As an educational tool, DynaLearn has been evaluated in multiple international educational settings and is considered motivating when it comes to learning by modeling (Bredeweg et al., 2013). Though DynaLearn is an effective educational tool, it supplies the learners with insufficient feedback regarding the resulting states. For instance, if the states that result from the simulation deviate from the learners’ expectations, the system supplies little explanation that aids the learners’ comprehension in these cases of discrepancy. Consequently, failure to understand how the states are generated is likely to result on the learners’ side. This may cause the learning process to stagnate, as the learner is unable to adequately develop an understanding of the subject matter (Beek & Bredeweg, 2012).

Research aimed at implementing feedback in DynaLearn has focused on the discrepancies between the actual and the learner-expected simulation results of a model, as well as identifying what aspects are accountable for these differences (Beek & Bredeweg, 2012). The referenced study works towards a program that allows the learner to state their expectations concerning the model’s behaviour. The program then automatically infers the mismatch between the learner’s expectation and the actual model and assists with discrepancy repair.

Other research focuses on providing semantic feedback, by comparing the learner’s model to the solution provided by the teacher (Lozano et al., 2015). Both studies show the added value of automated feedback. However, the program created by Beek & Bredeweg (2012) was not completed and Lozano et al.’s (2015) approach requires a norm model, which is not always possible or desirable.

1.3 The current research

From section 1.2 it follows that a feature that explains the occurrence of the resulting states based on the learner’s current model is needed. The research presented in this thesis aims to increase the learner’s comprehension of the workings of the underlying reasoning engine and to help construct the learner’s understanding of the subject matter. This uncovering of the hidden reasoning steps of reasoning systems has been proven to be increasingly important and has spurred the growth of fields like Explainable AI (Samek et al., 2017).

In order to design the explanation, how models are processed by learners is investigated in section 2.1. Then a distinction is made between three types of instruction (Merrill, 2012) in section 2.2, which are expanded to the four categories of explanations that are presented in section 3. By analyzing Liem’s
2 Theoretical Background

The general aim of the explanation is to improve learners’ understanding of their own model and the subject matter. To design such an explanation, the way models are interpreted according to cognitive psychology is researched. Section 2.1 explores how learners process a conceptual model by forming an equivalent representation in their mind. Additionally different types of explanation are investigated in section 2.2. The distinction is made between ‘kind-of’, ‘what-happens’ and ‘how-to’ explanations which forms the theoretical backbone for the different types of explanation that are implemented into DynaLearn. Lastly, common modeling errors in DynaLearn are discussed in section 2.3 to determine what the focus of the explanation should be to avoid learners making mistakes.

2.1 How learners process models

The explanations in DynaLearn should contribute to creating a correct mental model of the system that is being modeled. A mental model is an “internal conceptual representation of an external system whose structure resembles the perceived structure of that system” (Doyle & Ford, 1998). To design such an explanation, it is important to consider how mental models are generated. Firstly, it is discussed how information is processed and retained in the memory according to the information processing model. As DynaLearn is a multimedia application consisting of images and text, special attention is given to how the working memory processes text and images. Here, a distinction is made between descriptive and depictive representations.

2.1.1 The information processing model

The information processing model is a theory from cognitive psychology that postulates three types of memory: sensory, working and long-term memory (Khalil & Elkhider, 2016). Incoming information gets processed by these three types of memory. Information from the environment gets perceived by the sensory memory, which retains an exact copy of what is seen or heard (Huitt, 2003). This copy only lasts for a limited amount of time. If the learner pays enough attention to the information in the sensory memory, it gets passed to the working memory, which is where we consciously execute mental activities (Khalil & Elkhider, 2016). The capacity of these mental activities is limited, as the working memory has a relatively small capacity (Baddeley, 2013).

Compared to the sensory and working memory, the long-term memory has relatively permanent storage (Huitt, 2003). To move information from the working memory to the long-term memory, the learner needs to perform elaborative rehearsal (Khalil & Elkhider, 2016). Elaborative rehearsal is deep learning,
where the learner organizes the input information to have meaning and to create understanding. Another more shallow type of rehearsal is maintenance rehearsal. This means the learner remembers the information but does not process it on a deeper level (Khalil & Elkhider, 2016). When it comes to designing instructions the goal should be to encourage elaborative rehearsal over maintenance rehearsal.

2.1.2 Processing descriptive and depictive representations

The working memory can be split into two channels; one that processes depictive representation and one that processes descriptive representations (Schnotz & Bannert, 2003). Depictive representations remain close to the structure they are representing. They match the structural characteristics of what they are trying to depict and thus allow us to read relational information. Examples of depictive representations are photographs, sculptures or physical models.

Descriptive representations are ‘symbols’ that refer to an object, for example written text or mathematical equations (Schnotz & Bannert, 2003). These symbols have a specific structure which relates them to the content they represent. DynaLearn consists solely of descriptive representations. It features a set of symbols for elements such as entities, quantities and proportionalities that represent their supposed real-life counterparts through DynaLearn’s conventions. For example, the dark blue circle with four smaller white circles labeled ‘Beer’ (bear in English) as shown in figure 1 is a symbolic representation of bears as an entity. In this section we will only discuss how descriptive representations are processed to form a mental model. Information on how depictive representations are handled can be found in the article by Schnotz and Bannert (2003).

Firstly, the symbolic representations undergo sub-semantic processing resulting in a surface representation. This surface representation is then processed again semantically to form a propositional representation of the semantic content. For example, when processing figure 1 the configuration between habitat and predator is transformed into LivesIn(Predator, Habitat). Finally, from this propositional representation a mental model is constructed. During this step, a transition is made from a descriptive to a depictive representation. The propositional representation and the mental model interact continuously through model construction and inspection. This process is guided by cognitive schemata (Schnotz & Bannert, 2003). Figure 2 depicts this information processing model.

2.2 Types of explanation

Modeling is a complex task and as such requires a range of skills. Different kinds of skills require a different kind of instructional approach. Hence, in this research a distinction is made between three types of component skills. A component skill is “a combination of knowledge and skill that is required to solve a complex problem” (Merrill, 2012). For each type of component skill, a different kind of explanation is designed.

The Component Display Theory identifies five types of component skills; information-about, part-of, kind-of, what-happens and how-to skills (Merrill, 2012). Information-about deals with facts and associations, while part-of specifies the location of parts with regard to a whole object or system. Both have
Figure 2: The Information Processing Model (Khalil & Elkhider, 2016) with separate channels in the working memory for depictive and descriptive representations (Schnotz & Bannert, 2003). The black arrows represent processes (named by the label in the grey box) and the three darkening squares represent the three types of memory which lead to an increasingly deeper understanding. The circles represent the phases of the representation of the information.
a supportive role when it comes to learning and are often prerequisites for the other component skills. Thus, the focus is mainly on the remaining three component skills as they subsume these two component skills.

The goal of the **kind-of** instructional strategy is to “classify unencountered instances — objects, devices, procedures, actions or symbols — as belonging to a particular class.” (Merrill, 2012). When it comes to the presentation of the explanation it is important that the name of the class is discussed. Furthermore, the properties that are important for distinguishing between classes should be highlighted. The learner needs to be able to determine the class membership of a concept based on the values of the relevant properties. **Kind-of** instructions are associated with elaborating concepts and answering ‘what’-questions.

**What-happens** instructional strategies enable the learner to “given a set of conditions, predict the consequences for unencountered instances of the process. And given an unexpected consequence identify the missing or flawed conditions responsible for the consequence.”. This strategy explains a process. To this end, the conditions for each event in the process are given. **What-happens** is associated with ‘why’-questions.

The **how-to** instructional strategy aims to “perform a series of actions that lead to some desired consequence for unencountered instances of the task”. This strategy explains a procedure. This explanation should include an ordered set of steps that are required to successfully execute the procedure. **How-to** instructions are associated with giving step-by-step instructions and ‘how’-questions.

### 2.3 Modeling errors in DynaLearn

In general, modeling mistakes in DynaLearn can be divided into two categories; misconceptions concerning the formalisms of the reasoning engine and misconceptions about domain knowledge. Liem (2013) distinguished between these two categories by identifying formalism-based features and domain representation-based model features. Mistakes related to the formalism-based model features, for example, include inconsistencies that render the reasoning engine unable to make inferences. The other category relies on the human interpretation of the domain. For instance, a population model can be assessed in terms of the correct use of ‘number of deaths’: whether it is an entity or a quantity or if perhaps a more domain specific synonym should be used (for example ‘mortality’).

The focus of the explanation will be on the formalism-based features, as this requires no domain knowledge and will inform the user of the reasoning engine’s inferences in all contexts.

The current version of DynaLearn supports three modeling levels; standard, extended and extended+. The research presented in this thesis focuses on the standard modeling level. Liem (2013) offers a comprehensive list of 36 commonly occurring modeling errors in DynaLearn, of which 20 are applicable to the standard modeling level. Of these 20 modeling errors, 8 concern the formalism-based features of the model. A short description of these 8 modeling errors is given in table 1.

These modeling errors are all related to either the organization of proportionalities, the derivative values that are set (or unset) and inconsistencies or underspecified values that lead to unwanted or missing states. In DynaLearn’s standard level these unwanted or missing states are always due to proportional-
ities and underspecified values. The design of the explanation will have a high focus on proportionalities and derivative values, as these elements are prone to formalism-based modeling errors.

Table 1: Formalism-based modeling errors of the standard modeling level discussed in Liem (2013).

<table>
<thead>
<tr>
<th>Number</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Loop of proportionalities</td>
</tr>
<tr>
<td>27</td>
<td>Value assignments on derivatives</td>
</tr>
<tr>
<td>30</td>
<td>Non-firing model fragments</td>
</tr>
<tr>
<td>32</td>
<td>Unknown quantity values in simulations</td>
</tr>
<tr>
<td>33</td>
<td>Simulation of scenario produces no states</td>
</tr>
<tr>
<td>34</td>
<td>Dead-ends in state graph</td>
</tr>
<tr>
<td>35</td>
<td>Missing required state in state graph</td>
</tr>
<tr>
<td>36</td>
<td>Incorrect state in state graph</td>
</tr>
</tbody>
</table>

3 Designing and implementing

Based on the distinction identified in section 2.2, a set of different types of explanation are designed (what, why, why not and how) and integrated into DynaLearn’s front-end. The explanations are designed to cover all aspects of the standard level and a smaller portion of extended and extended+.

In section 3.1 the general design choices that apply to all explanations are discussed first. Secondly, how and when the different explanations can be prompted by the learner is considered (section 3.2). Section 3.3, 3.4, 3.5 and 3.6 elaborate how each explanation generates its content and examples of how the explanation looks are given. Lastly, in section 3.7 attention is paid to the implementation of the explanations and pseudo-code is shown.

3.1 Design Choices

The aim is to seamlessly integrate explanations into the DynaLearn interface. DynaLearn’s front-end is written in AngularJS, as are the explanations. To make the help as intuitive as possible, only one button was added, which serves to enable the ‘what’-explanations. All other explanations follow from (right)-clicking on the appropriate elements at the appropriate time.

For displaying the textual hints the qTip\(^2\) package\(^1\) was used. The style of the qTips was matched to that of DynaLearn, to appear like a natural extension of the program. The qTips are ideal for the purpose of these explanations, as they are easy to customize when it comes to text and display and hide functions. Other than that, qTips are shaped like speech bubbles and thus help the learner infer which element of the model the explanation refers to by following the tail of the speech bubble.

Though all types of explanation were designed individually while acknowledging their respective functions, the following design choices apply to all types.

\(^1\)http://qtip2.com/
• When referring to a quantity in an explanation, the corresponding entity is always mentioned. This is because the same quantity name can be used for different entities when modeling (for example using multiple instances of the quantity ‘size’ when creating a model of an ecosystem with various inhabitants). To distinguish between these quantities, it is vital to mention the associated entity.

• All elements of a model that are mentioned in an explanation are in bold, so the learner’s attention is drawn to the elements of the model that are affecting each other. Using such signals to highlight important concepts increases learning (Renkl & Scheiter, 2017; Mayer & Moreno, 2003).

• As excluding extraneous materials increases learning (Clark & Mayer, 2016; Mayer & Moreno, 2003), the text is kept as short and to the point as possible.

• The tone of the explanation is kept as informal as possible, as this has been proven to increase learning (Clark & Mayer, 2016).

3.2 Prompting the explanations

In the current section the mechanisms behind prompting the explanations are discussed. A summary of when and how the explanations can be prompted by the learner is given in table 2 and 3.

At any given time the learner can prompt a ‘what’-explanation by enabling the ‘what’-explanations (by pressing the button in figure 3) and hovering over (multiple) elements. The explanation disappears once the learner’s mouse leaves the element. A pilot version of the ‘what’-explanation showed the definition when hovering over an element without requiring a button to switch this feature on. However, this mechanism interfered with the modeling, as speech bubbles would unwillingly show up when connecting elements and would often cover up buttons.

Figure 3: The button that enables the ‘what’-explanation. The image on the right-hand side shows the button once it is activated.

A ‘why’-explanation can only be triggered after a successful simulation has been run. The simulation results are marked by arrows (see the green arrows in figure 1). When the user clicks an arrow, its value is explained in terms of the conditions that led to it. The learner can cycle through the arrows from left to right (providing a comprehensive explanation of the entire model), or click on them in any other desired order. When the user clicks a second time the explanation is hidden.

Instead of offering an explanation for the entire model at once, a single simulation result is explained by highlighting the direct cause of the derivative value. The information is given in chunks (divided by each green arrow) as to not overload the learner and improve learning (Chandler & Sweller, 1991). Furthermore,
the learner can control the speed of the explanation, as they control clicking on the arrows which improves learning results (Clark & Mayer, 2016).

The ‘why not’-explanation is triggered everytime a simulation is run. When inconsistencies or underspecified values are found, they are automatically reported to the learner. When the learner clicks anywhere on the screen the explanation disappears.

After a simulation has been run the learner can request a ‘how’-explanation by right-clicking on a derivative value. The explanation will disappear once the user (left-)clicks.

Table 2: Summary of how the learner can show and hide the explanations and what they cover.

<table>
<thead>
<tr>
<th>Explanation</th>
<th>Show</th>
<th>Hide</th>
<th>Covers</th>
</tr>
</thead>
<tbody>
<tr>
<td>What</td>
<td>When hovering over an element</td>
<td>When no longer hovering over element</td>
<td>Definitions of model ingredients</td>
</tr>
<tr>
<td>Why</td>
<td>When clicking a simulation value</td>
<td>On click</td>
<td>Whether the derivative value is set by the user or a result of proportionalities</td>
</tr>
<tr>
<td>Why not</td>
<td>After prompting a simulation on an incorrect model</td>
<td>On click</td>
<td>Whether there are underspecified values or inconsistencies</td>
</tr>
<tr>
<td>How</td>
<td>When right-clicking a derivative value</td>
<td>On click</td>
<td>How to achieve the desired derivative value by changing other derivative values</td>
</tr>
</tbody>
</table>

Table 3: Summary of when which explanations can be triggered.

<table>
<thead>
<tr>
<th>DynaLearn phases</th>
<th>Available explanations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before simulation</td>
<td>What</td>
</tr>
<tr>
<td>During simulation</td>
<td>What</td>
</tr>
<tr>
<td>After simulation</td>
<td>Why</td>
</tr>
<tr>
<td></td>
<td>Why not</td>
</tr>
<tr>
<td></td>
<td>How</td>
</tr>
</tbody>
</table>

3.3 What

The ‘what’-explanation corresponds with the kind-of instructions (section 2.2). Kind-of instructions are associated with answering ‘what’-questions (i.e. What is an entity?). Therefor, the kind-of instruction supplies information about the ingredients of a model.

The ‘what’-explanation supplies definitions of all model ingredients included in the standard model level. The definitions are taken from the DynaLearn glossary\(^2\) and are shown in table 4.

When the cursor focuses on an element, the corresponding definition will pop up next to it (figure 4). This mechanism is equivalent for every other type of element, while every textual explanation is matched to the element.

\(^2\)https://ivi.fnwi.uva.nl/tcs/QRgroup/DynaLearn/glossary/
3.4 Why

The ‘why’-explanation corresponds with the what-happens instructions (section 2.2). What-happens is associated with ‘why’-questions (i.e. Why is this quantity increasing?). In DynaLearn this means that the conditions that are responsible for a specific simulation result should be explained.

When the learner clicks a simulation result, the algorithm that determines why the simulation result has its specific value is triggered. This algorithm checks for 3 conditions (figure 9) which, depending on the specific conditions, result in one of four possible scenarios.

1. **Scenario 1:** The learner set the value (figure 5).
2. **Scenario 2:** A single proportionality influences the quantity (figure 5).
3. **Scenario 3:** Multiple unambiguous proportionalities influence the quantity (figure 6).
4. **Scenario 4:** Multiple ambiguous proportionalities influence the quantity (figure 7 and 8).

Motivated by the psychology behind processing depictive representations discussed in section 2.1.2, the ‘why’-explanation focuses on the propositional representation of the model by highlighting the two subjects and their relation in each explanation (except scenario 1). For example, the explanation depicted in figure 5 on the right is essentially the text form of its propositional representation \textit{Influences(Amount of Bear, Amount of Fish)}. This way the explanation contributes to forming a correct mental model, as learners process a model through converting it to a propositional representation (Schnotz & Bannert, 2003).

The distinction between a quantity that is being influenced by a single proportionality or multiple proportionalities is made to ensure learners are aware of the multiple influences (by generating a message that states this explicitly). Ambiguous and unambiguous proportionalities also generate separate explanations to emphasize that the model is ambiguous. This reduces the chance of learners getting confused by the generation of multiple states in an ambiguous situation.
<table>
<thead>
<tr>
<th>Element</th>
<th>Glossary definition</th>
<th>Dutch Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Entities are the physical objects or abstract concepts that play a role within the system.</td>
<td>Entiteiten zijn fysische objecten of abstracte concepten die een rol spelen in het systeem.</td>
</tr>
<tr>
<td>Quantity</td>
<td>Quantities represent changeable features of entities and agents. Each quantity is associated with a derivative that can be either decreasing ('min'), steady ('zero') or increasing ('plus').</td>
<td>Hoeveelheden vertegenwoordigen veranderlijke kenmerken van entiteiten. Elke hoeveelheid heeft een afgeleide die aangeeft of de hoeveelheid afneemt ('min'), stabiliseert ('nul') of toeneemt ('plus').</td>
</tr>
<tr>
<td>Proportionality</td>
<td>Proportionalities are directed relations between two quantities. They propagate the effects of a process, i.e. they set the derivative of the target quantity depending on the derivative of the source quantity. For this reason, they are also referred to as indirect influences. Like influences, proportionalities are either positive or negative. A proportionality $P^+(Q2, Q1)$ causes $Q2$ to increase if $Q1$ increases, decrease if $Q1$ decreases, and remain stable if $Q1$ remains stable (given that there are no other causal influences on $Q2$). For a proportionality $P^-$ this is just the opposite.</td>
<td>Proportionalities are always either negative or positive, so there is no general definition needed.</td>
</tr>
<tr>
<td>Negative</td>
<td>Not provided</td>
<td>Een proportionaliteit geeft de relaties tussen twee hoeveelheden aan. Via de proportionaliteit wordt de richting van de verandering doorgegeven. Deze proportionaliteit is negatief en dit betekent dat de verandering in tegenovergestelde richting wordt doorgegeven. Als de bronhoeveelheid toeneemt, dan neemt de doelhoeveelheid af en vice versa. Als de bronhoeveelheid stabiliseert doet de doelhoeveelheid dat ook.</td>
</tr>
<tr>
<td>Positive</td>
<td>Not provided</td>
<td>Een proportionaliteit geeft de relaties tussen twee hoeveelheden aan. Via de proportionaliteit wordt de richting van de verandering doorgegeven. Deze proportionaliteit is positief en dit betekent dat als de bronhoeveelheid toeneemt, afneemt of stabiliseert, de doelhoeveelheid dat ook doet.</td>
</tr>
<tr>
<td>Derivative</td>
<td>The derivative indicates the direction of change of a quantity. This can be either 'min' (decreasing), 'zero' (stable), or 'plus' (increasing).</td>
<td>De afgeleide geeft de richting van de verandering van een hoeveelheid aan. Deze afgeleide is positief teken dat als de bronhoeveelheid toeneemt, afneemt of stabiliseert, de doelhoeveelheid dat ook doet.</td>
</tr>
<tr>
<td>Negative</td>
<td>Not provided</td>
<td>De afgeleide geeft de richting van de verandering van een hoeveelheid aan. Deze afgeleide is negatief teken dat als de bronhoeveelheid toeneemt, afneemt of stabiliseert, de doelhoeveelheid dat ook doet.</td>
</tr>
<tr>
<td>Positive</td>
<td>Not provided</td>
<td>De afgeleide geeft de richting van de verandering van een hoeveelheid aan. Wanneer deze waarde ('min') is geselecteerd, neemt de hoeveelheid af.</td>
</tr>
<tr>
<td>Derivative</td>
<td>Not provided</td>
<td>De afgeleide geeft de richting van de verandering van een hoeveelheid aan. Wanneer deze waarde ('nul') is geselecteerd, is de hoeveelheid stabiel.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Configurations are used to model relations between instances of entities and agents. Configurations are sometimes referred to as structural relations.</td>
<td>Configuraties worden gebruikt om de relaties tussen entiteiten aan te geven.</td>
</tr>
</tbody>
</table>
Figure 5:
**Left**: ‘Why’-explanation - Scenario 1: An example of a reminder that the learner set the value.
**Right**: ‘Why’-explanation - Scenario 2: An example of the explanation if there is a single proportionality.

Figure 6: ‘Why’-explanation - Scenario 3: An example of the explanation if there are multiple unambiguous proportionalities.
Figure 7: ‘Why’-explanation - Scenario 4: An example of the explanation if there are multiple ambiguous proportionalities and the increasing state is selected.

Figure 8: ‘Why’-explanation - Scenario 4: An example of the explanation if there are multiple ambiguous proportionalities and the stable state is selected.
3.5 Why Not

The ‘why not’-explanation is the counterpart of the ‘why’-explanation that answers the question ‘Why is the simulation not running?’. In the standard modeling level there can be two reasons for this; the model either contains underspecified values (figure 10) or inconsistent derivative values (figure 11).

The ‘why not’-algorithm will collect all the quantities that are underspecified and all the proportionalities that lead to an inconsistent derivative value. To catch all underspecified values, the algorithm cycles over each proportionality. Each proportionality has a from-quantity and a to-quantity (the quantity the proportionality points from and the quantity that the proportionality points to respectively). Essentially, the rule is that all proportionalities that have no proportionality pointing to them need to be set (or equivalently, each proportionality that forms the beginning of a proportionality chain). This is because if a proportionality (p1) does have a proportionality (p2) that points to them, it can derive the derivative value from the from-quantity of this proportionality (p2) (assuming the from-quantity has a set value).

At the same time all the inconsistent proportionalities are found while cycling the proportionalities. A proportionality can only be inconsistent if both the from-quantity and the to-quantity are set (by the user or as a result of a simulation) and there are no ambiguous influences. A chain of proportionalities can be consistent up until the last proportionality, thus the entire chain needs to be checked recursively (see section 3.7 for more information on the function responsible for this). A derivative value with ambiguous influences can lead to a plus, zero and min derivative value and thus cannot be inconsistent. The ‘why not’-algorithm is visualized in figure 12.

Figure 9: The algorithm that determines the ‘why’-explanation scenario.
Figure 10: An example of the ‘Why not’-explanation shown for an underspecified value.

Figure 11: An example of the ‘Why not’-explanation shown for an inconsistency.
3.6 How

The ‘how’-explanations matches with the how-to instructions (section 2.2). They describe a procedure (i.e. How do I get this quantity to increase?). In this implementation, the ‘how’-explanation lets the learner right-click on a derivative value and will demonstrate what quantities need to be set to what values to achieve the desired derivative value.

There are multiple ways a certain derivative value can be achieved as a simulation result. The ‘how’-explanation prompts the simplest and least intrusive method to the learner. The explanation assumes the structure of the model will remain the same and only gives suggestions as to what derivative values need to be set to attain the desired effect. It also assumes the learner wants the desired derivative value to be unambiguous (resulting in a single state, where all proportionalities have the same effect on the derivative value).

There are two possible scenarios; no proportionalities are pointing to the derivative value or there are proportionalities pointing to the derivative value (figure 15). If there are no proportionalities pointing to the derivative value, then the only way to achieve the desired value is to set it manually (figure 13).

If there are proportionalities pointing to the derivative value, a function is activated that recursively finds the proportionality at the end of each chain of proportionalities (see section 3.7 for the pseudo-code). This function terminates
at the end of a proportionality chain and will generate a message that shows what value the attached quantity needs to be set to (figure 14).

Figure 13: ‘How’-explanation - Scenario 1: An example of the explanation if there are no proportionalities pointing to the derivative value.

Figure 14: ‘How’-explanation - Scenario 2: An example of the ‘how’-explanation if there are (two) proportionalities pointing to the derivative value.
3.7 Implementation

The explanation is embedded entirely into the canvas controller of DynaLearn. This canvas controller is divided into a model-view controller and a simulation-view controller. These controllers regulate the view of the screen during modeling and simulating respectively. The Cytoscape.js package$^3$ is used for creating the model. Each model-element has properties attached to it that describe its relative place and function in the model (figure 5).

Table 5: Overview of which properties are stored for which elements.

<table>
<thead>
<tr>
<th>Property</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>All</td>
</tr>
<tr>
<td>type</td>
<td>All</td>
</tr>
<tr>
<td>nodename</td>
<td>All</td>
</tr>
<tr>
<td>parentId</td>
<td>All</td>
</tr>
<tr>
<td>childIds</td>
<td>All</td>
</tr>
<tr>
<td>derivative to</td>
<td>Derivative values</td>
</tr>
<tr>
<td>from</td>
<td>Proportionalities</td>
</tr>
</tbody>
</table>

The implementation of the algorithms revolve around these properties. For example, in the algorithm of the ‘why not’-explanation (figure 12), checking if the from-quantity is set means checking if a proportionality has a value for the from-property.

Checking for ambiguous influences becomes collecting all the proportionalities that are pointed to a quantity and determining whether they have a positive (positive proportionality and plus derivative value or negative proportionality and minus derivative value), neutral (a zero derivative value) or negative (positive proportionality and minus derivative value or negative proportionality and a plus derivative value) influence. If there are proportionalities that have both a positive and a negative influence, it can be concluded that the influences are ambiguous.

$^3$https://github.com/cytoscape/cytoscape.js
Both the ‘why not’-explanation and the ‘how’-explanation use recursion to check all the proportionality chains that can possibly lead from one proportionality. The ‘why not’-explanation does this to ensure that the entire chain is consistent and the ‘how’-explanation ends up reporting the value that needs to be set to the user at the very start of the chain to ensure the right result at the end of the chain. The pseudo code of this recursive function (of the ‘how’-explanation) is given below. The function calculates the necessary value needed to achieve the desired effect. If the function encounters a negative proportionality and the desired effect is an increase, the value gets inverted and a decrease is suggested. The opposite occurs if the algorithm finds a positive proportionality, then for every decrease a decrease is suggested and for every increase an increase. If the learner wants a stable value all proportionalities that point to the derivative value need to be set to zero (to achieve an unambiguous stable derivative value).

Function $\text{DDV}(\text{quantity.id, original derivative value})$:

```
props = all proportionalities that have quantity.id as the to-property;
for proportionality in props do
  if proportionality.type is negative then
    if derivative value is plus then
      DDV(proportionality.from, minus derivative value);
    else if derivative value is minus then
      DDV(proportionality.from, plus derivative value);
    else
      DDV(proportionality.from, original derivative value);
  end
else
  DDV(proportionality.from, original derivative value);
end
generate the explanation
```

Algorithm 1: Pseudo-code of how the ‘how’-algorithm traverses the chains of proportionalities to determine the correct derivative value. The function is called DDV (Determine Derivative Value).

4 Evaluation

4.1 Technical evaluation and scope

To determine the scope of the explanation a comprehensive list of modeling errors supplied by Liem (2013) is examined. Based on this list, it is determined how many modeling errors can be addressed by the explanations implemented in this research.

For the technical evaluation the basic chunks of a model on the standard modeling level are identified. It is checked whether the explanations cover these basic model components to determine if the explanations supply a full coverage of all possible modeling scenarios.
4.2 Classroom evaluation

As DynaLearn is designed to be used in an educational context (Bredeweg et al., 2013), the effectiveness of the explanations was evaluated with a classroom experiment. Twenty-two students in 5VWO of a high school carried out a set of assignments designed to make them interact with the explanations. The three assignments can be found in the appendix (section 8.1). All students were assigned 30 minutes to work on the assignments and were asked to fill out a survey afterwards (see appendix, section 8.2). The help-seeking behaviour of the students in DynaLearn was documented. To this extend, the amount of times the help functions were called were tracked together with the modeling actions performed by the students (such as remove entity, create proportionality, etc.). None of the students had any previous experience with DynaLearn and were thus given a manual (2 pages), which focused purely on explaining the functions of each button in DynaLearn and how the explanation can be triggered. The manual did not contain any references to model elements or their definitions, as to not interfere with the explanations.

The distribution of the different type of profiles is summarized in table 6. To account for the differentiating subject profiles of the students, the assignments featured subjects that tailor to all profiles (assignment 1 has an economic undertone, assignment 2 is more general and assignment 3 appeals to the ‘Nature’-profiles). Another reason for creating three assignments was to measure whether the use of the help functions would decrease when the student used them on a model before and is now more familiar with the concepts.

Table 6: Profile distribution of students

<table>
<thead>
<tr>
<th>Profile</th>
<th>Amount of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nature &amp; Technique</td>
<td>5</td>
</tr>
<tr>
<td>Nature &amp; Health</td>
<td>8</td>
</tr>
<tr>
<td>Economy &amp; Society</td>
<td>8</td>
</tr>
<tr>
<td>Culture &amp; Society</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>22</strong></td>
</tr>
</tbody>
</table>

5 Results

5.1 Scope of the explanation

As mentioned in section 2.3, Liem (2013) features a comprehensive list of modeling errors in DynaLearn that points out 8 formalism-based errors when using the standard modeling level. Table 7 lists which explanations cover which modeling errors. The explanations implemented in this thesis cover 8/8 modeling errors of the standard modeling level. Consequently, approximately 8/14 of modeling errors of the extended and extended+ modeling level are covered by this research’ implementation. It should be noted that not all explanations translate to the extended and extended+ modeling level. For example, what would be considered an underspecified value in the standard modeling level is often acceptable in the extended and extended+ modeling levels, as quantity values can make up for unset derivative values. The explanations should be re-evaluated.
before being applied to the extended and extended+ modeling levels.

Error 22 is corrected by the ‘why’ and ‘why not’-explanation. A loop of
proportionalities should be avoided in DynaLearn. Whenever a loop of propor-
tionalities occurs with an unset value, the ‘why not’-explanation will commu-
nicate this to the learner. If the value is set, then the ‘why’-explanation will
help the learner understand the resulting simulation and possibly prompt the
learner to break the loop of proportionalities. If there are value assignments
on derivatives that are inconsistent (error 27), then the ‘why not’-explanation
will notify the learner of this. Non-firing model fragments, because the correct
conditions are not fulfilled (error 30), is helped by the ‘how’-explanation. Er-
ror 32 (where there are unknown quantity/derivative values in the simulation
that should be set are) is brought to the learner’s attention via the ‘why not’
explanation. Errors 33, 34 and 35 can refer to no states, dead-ends or miss-
ing required states due to inconsistencies derivative values or the model being
internally inconsistent (Liem et al., 2013). If no derivative values are set and
the expected states are still not generated, the model is internally inconsistent.
When error 33, 34, or 35 occurs due to an internal modeling error it is covered
by the ‘why’-explanation, else it is covered by the ‘why not’-explanation. Error
36 refers to incorrect states in the state graph (for example, the unintentional
generation of three state because of ambiguity) and is brought to the learner’s
attention through the ‘why’-explanation.

Table 7: Summary of which explanations cover which modeling errors.

<table>
<thead>
<tr>
<th>Number</th>
<th>Short description</th>
<th>Covered by</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Loop of proportionalities</td>
<td>Why</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Why not</td>
</tr>
<tr>
<td>27</td>
<td>Value assignments on derivatives</td>
<td>Why</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Why not</td>
</tr>
<tr>
<td>30</td>
<td>Non-firing model fragments</td>
<td>How</td>
</tr>
<tr>
<td>32</td>
<td>Unknown quantity values in simulations</td>
<td>Why</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Why not</td>
</tr>
<tr>
<td>33*</td>
<td>Simulation of scenario produces no states</td>
<td>Why</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Why not</td>
</tr>
<tr>
<td>34*</td>
<td>Dead-ends in state graph</td>
<td>Why</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Why not</td>
</tr>
<tr>
<td>35*</td>
<td>Missing required state in state graph</td>
<td>Why</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Why not</td>
</tr>
<tr>
<td>36</td>
<td>Incorrect state in state graph</td>
<td>Why</td>
</tr>
</tbody>
</table>

* ‘Why’ if the model is internally consistent else ‘why not’.

5.2 Technical evaluation

The ‘what’-explanation covers each model element available in the standard
modeling mode and thus provides full coverage. The ‘why’-explanation gener-
ates the explanation based on the proportionalities that point to clicked deriva-
tive value directly. This means the only possible scenarios are the ones men-
tioned in section 3.4. Consequently, the ‘why’-explanation also provides full cov-
erage of all possible modeling scenarios. The ‘why not’ and ‘how’-explanations
are trickier, however. A chain of proportionalities can be consistent up until the
last proportionality and the algorithm still needs to be able to detect this. Fur-
thermore, the derivative value that the user wants as a simulation result can be
influenced by multiple proportionalities. The proportionalities can, for example,
start out as a single chain but end up branching out, creating an intricate web of
proportionalities. This is why both these explanations use a recursive function
to detect the inconsistent derivative values or the derivative values that need to
be set to achieve the desired result. This function is explained in 3.7 and covers
all possible combinations of any number of proportionalities. This means all explanations provide a full coverage of every possible modeling scenario.

5.3 Classroom evaluation

5.3.1 Help-seeking behavior

A portion of the students did not manage to fulfill all the assignments in the allocated time. Thus only the help-seeking behavior of the first two assignments (or equivalently the first seven questions) are used for further analysis ($n = 13$). Data of students who did not get past the first assignment was discarded.

The total amount of calls to the explanations per student are summarized in figure 16. In the first assignment only the ‘what’ and the ‘why not’-explanation was used. The amount of calls to the ‘what’-explanation is possibly over-exaggerated, which is explained in more detail in section 5.3.3. Both low and high scoring students show diversity in terms of how frequently the explanations were used. This suggests there is no strong correlation between a student’s score and how often they use the explanations. How often a student triggers the explanation may be more dependent on other factors, such as how much the student trusts their own judgment. Another interesting observation is that the use of the explanations tends to increase among the high-scoring students when comparing the first assignment to the second. This may be because the students are more experienced with triggering the explanations when using them in the second assignment. For example, leading them to the discovery of the ‘why’-explanation. The lower-scoring students used the explanations a lot less frequently during the second assignment, which possibly indicates they were more familiar with the definitions and causes of inconsistencies that they learned about through the explanation in the first assignment or that they deemed the explanations as unhelpful. The ‘how’-explanation was used by some of the students that did not manage to complete the first two assignments. However, it is clearly used a lot less frequently than the other explanations.

Students scored the best on questions that concerned the definitions of elements (1.1 & 2.1), with a 100.0% of students answering correctly. This is reflected by students using the ‘what’-explanation significantly more frequently than the others. On average a question was answered correctly 62.6% of the time. The question that led to the most incorrect answers was 2.4, which required the student to examine an ambiguous simulation result. Only 30.8% of students had the right answer. The ‘why’-explanation especially elaborates on how ambiguity is treated in DynaLearn and the fact that students rarely used the ‘why’-explanation may have contributed to question 2.4 being considered a difficult question.

5.3.2 Survey

The survey consists of four parts. In the first part background information about the student is gathered, namely what their subjects profile is and if they have experience with programs similar to DynaLearn. The next three parts concern the explanations and prompt the student to judge the display, content and effectiveness of the explanations. This survey uses a 5-point Likert scale, as they reduce frustration among respondents and have been shown to lead
Figure 16: Calls to the explanations with respect to the individual total score of each student. One bar represents one student. Students with the same total score are grouped by the vertical black lines. The group on the far right has a perfect score (7/7).
to equivalent results compared to 7-point Likert scales once re-scaled (Dawes, 2008).

The students reported to have very little experience with creating diagrams and using programs like DynaLearn ($\mu = 1.6$ and $\mu = 2.0$; table 8). This may have contributed to most students being unable to finish all assignments and having difficulties with using DynaLearn (see also section 5.3.1 and 5.3.3).

In general the students felt neutral about their experience with the explanations in DynaLearn, which is exemplified by the fact that the average score given by the students for all questions is 3.0 (table 8). However, the students felt relatively strong about the assignments being impossible without the explanation ($\mu = 3.8$). This emphasizes the fact that students have trouble grasping the concepts in DynaLearn without any explanations and that they should be developed further.

Additionally, all students were asked whether they had any remarks concerning the display, content and effectiveness of the explanations. Concerning the display of the explanation one student mentioned that examples and pictures would improve the explanation. Other than that students seemed to be satisfied with the display of the explanation.

When it comes to the content of the explanation seven students reported that they would like more general help that focuses on how DynaLearn works (in the sense of the interface and how to add, remove and modify model elements etc.). Student’s remarks are translated from Dutch. One of these students said “Explain things more specifically for beginners. People like me who have no experience with programs like this, have no clue what they are doing”. Another said “You should really have more experience with the program and subjects to really understand it”. Explaining how the DynaLearn interface works is out of the scope of the explanation developed in this thesis. But the need for a more rudimentary explanation focused on familiarizing a beginner with the operations in DynaLearn should be noted for future research. Another remark that was made concerning the content of the explanation was that the explanation should also demonstrate how to improve modeling mistakes. One of the two students who made this comment said “The explanation should not just explain what’s wrong, but also how it can be improved”. The students expressed a need for an extension of the current ‘how’-explanation, which now only covers how to achieve certain derivative values. Another student said “the definitions are superfluous”. This refers to the ‘what’-explanation. One students also mentioned that it was unclear how the proportionalities functioned.

At the end of the survey the students were given the opportunity to give general remarks about DynaLearn. This showed the program had a mixed reception. One student reported “It was complicated”, another said “Fun program”.

<table>
<thead>
<tr>
<th>Translated Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>$\mu$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I often use the computer to learn</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>13</td>
<td>4</td>
<td>4.0</td>
<td>0.7</td>
</tr>
<tr>
<td>I often create diagrams on the computer</td>
<td>13</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1.6</td>
<td>0.9</td>
</tr>
<tr>
<td>It’s easy for me to use a program like DynaLearn</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>2.0</td>
<td>1.1</td>
</tr>
<tr>
<td>Total</td>
<td>23</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>4</td>
<td>2.5</td>
<td>1.4</td>
</tr>
</tbody>
</table>

27
Table 8: Survey results of the questions regarding the students’ experience with programs similar to DynaLearn and the explanation (n = 22).

<table>
<thead>
<tr>
<th>Translated Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>μ</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>The explanations are easy to use</td>
<td>3</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>2.5</td>
<td>1.0</td>
</tr>
<tr>
<td>The activation of the explanations make sense</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>11</td>
<td>0</td>
<td>3.0</td>
<td>1.1</td>
</tr>
<tr>
<td>The display of the explanations make sense</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>2.6</td>
<td>1.0</td>
</tr>
<tr>
<td>The explanations are easy to follow</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td>2.6</td>
<td>0.8</td>
</tr>
<tr>
<td>The explanations are complete</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>1</td>
<td>3</td>
<td>1.1</td>
</tr>
<tr>
<td>None of the explanations contain superfluous text</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>14</td>
<td>1</td>
<td>3.5</td>
<td>0.9</td>
</tr>
<tr>
<td>Without the explanations the assignments are impossible</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>3.8</td>
<td>1.3</td>
</tr>
<tr>
<td>The explanations help me understand what the elements in DynaLearn represent</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>10</td>
<td>0</td>
<td>3.0</td>
<td>1.1</td>
</tr>
<tr>
<td>The explanations help me understand how DynaLearn’s elements influence each other</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>0</td>
<td>2.9</td>
<td>1.1</td>
</tr>
<tr>
<td>The explanations help me avoid making modeling mistakes</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>2.8</td>
<td>2.2</td>
</tr>
<tr>
<td>The explanations help me understand the difference between a complementary and substitute good</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>3.0</td>
<td>1.3</td>
</tr>
<tr>
<td>The explanations help me understand how self-driving cars affect organ donations</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>6</td>
<td>1</td>
<td>3.0</td>
<td>1.1</td>
</tr>
<tr>
<td>The explanations help me understand how water affects the habitable zone of a planet</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>2.8</td>
<td>1.3</td>
</tr>
<tr>
<td>Total</td>
<td>36</td>
<td>64</td>
<td>73</td>
<td>98</td>
<td>15</td>
<td>3.0</td>
<td>1.2</td>
</tr>
</tbody>
</table>

5.3.3 Informal Observations

Not all students managed to stay focused during the 30 minutes that they were supposed to make the exercises. Mainly because they had trouble navigating DynaLearn’s interface. However, familiarizing learners with how to navigate DynaLearn’s interface is beyond the scope of the explanation in this thesis. Other students collaborated on the exercises by exchanging tips on how to add, remove or modify model elements.

A reoccurring theme among the students was that they interpreted the proportionalities in the wrong way. A negative proportionality was seen as a decrease and a positive proportionality as an increase. For example, the negative proportionality between Netflix and HBO’s in the first exercise was often in-
terpreted as ‘If the demand for Netflix makes the demand for HBO decrease’. While the assignment required the students to set the demand of HBO to increase by changing the derivative value, some changed the proportionality to be positive (even though the assignment explicitly states this is not how it should be solved) to signify that the demand for Netflix makes the demand for HBO increase. This misunderstanding prevented some students from using the derivative values correctly.

Another reoccurring theme was that students forgot to turn the ‘what’-explanation off before performing a modeling action. This would lead to students accidentally triggering ‘what’-explanations and failing to add a model element because the explanation kept them from pressing the needed buttons. This is a possibly explanation for the relatively high frequency of ‘what’-explanations compared to the other explanations in section 5.3.1.

6 Discussion

The explanations deal with a large range of DynaLearn-scenarios which is emphasized in the technical evaluation (section 5.2). However, the ‘how’-explanation does not encompass all the ways in which a simulation result can be achieved. The explanation will prompt the least intrusive way to achieve the simulation result (by explaining which derivative values need to be set to what value), other possibilities would be changing the structure of the model by rearranging proportionalities or overriding the derivative value by manually setting a derivative to the desired value. In the future, the explanations can be expanded by suggesting multiple options that lead to the queried simulation result, so the learner can choose which option to execute. Additionally, as shown in the survey responses, the coverage of the ‘how’-explanation should be extended to handle questions such as ‘how do I resolve this inconsistency?’.

All possible model configurations of the standard modeling level and a portion of the extended and extended+ modeling levels are covered by the explanations. The extended levels consist of more elements and in general require a more elaborate explanation. These levels also provide more opportunities for the ‘how’-explanation, as they offer a way to solve ambiguity by specifying which proportionalities have a stronger influence than others by using inequalities. This makes way for answering the learner who wonders ‘How do I disambiguate the effect of the proportionalities on this quantity?’. In the future, the explanations can be extended to cover these levels as well.

This thesis focused solely on covering the formalism based features and has managed to do so successfully. Domain representation-based model features are nevertheless an interesting consideration for future research. This research should include collaborations with domain-experts to expand the explanation to elaborate on domain specific features, such as suggesting an entity name that is more suitable for the specific field. The research mentioned in the introduction by Lozano et al. (2015) has already made the first steps towards supplying such feedback for entity names, by doing a syntactic analysis and linking DynaLearn to DBpedia.

Some remarks about the classroom evaluation can also be made. Ideally the explanations would have been tested on learners with enough experience with DynaLearn’s interface or with more allocated time before starting the assign-
ment in which the students would get a thorough demonstration of how to use DynaLearn. Neither were available for this research. The fact that the students were beginners when it comes to DynaLearn also resulted in most students being unable to finish all three assignments, which hampered a statistical analysis of the help-seeking behaviour due to the small sample. The effect of the explanations should be measured with a larger sample and, perhaps, compared to a control group without built-in explanations in future research.

The activation of the explanations leaves room for improvement. The ‘what’-explanation should not be turned on via a button, as learners often forget to turn it off when no longer needed. A suggestion is that a small question-mark icon is added to the top right of each element. When the learner hovers over this question-mark for a few seconds the ‘what’-explanation is shown. This same mechanism can be repeated for the ‘why’, ‘why-not’ and ‘how’-explanations (by placing the same icon to the top right of derivative values and simulation results). This would also make the presence of the explanations more obvious to the learners and would urge learners to use it more than they did during the classroom evaluation.

7 Conclusion

Based on cognitive psychology and instructional design research, four different types of explanation are identified and applied to DynaLearn. These explanations offer help when fixing all common formalism-based modeling errors and cover all model configurations of the standard modeling level. During the classroom evaluation the majority of students reported that the explanations were essential for completing the assignments and that the display of the explanations was satisfactory. This research managed to cover all scenarios that it set out to cover through the explanations. Yet, there is enough uncharted territory left for explanations developed in future research to cover.

This thesis has demonstrated the significance of having help which is visual at all times and does not interfere with performing the modeling actions. The subdivision of the explanation into ‘what’, ‘why’, ‘why not’ and ‘how’ has shown to be promising, granted the ‘how’-explanation can be extended to contribute to the learning process even more. There is room to further improve and extend the explanations presented in this thesis, but they form a technologically solid and practically sufficient start towards helping the learner understand the underlying reasoning engine and the subject matter.
References


Dawes, J. (2008). Do data characteristics change according to the number of scale points used? an experiment using 5-point, 7-point and 10-point scales. *International journal of market research, 50*(1), 61–104.


8 Appendix
8.1 Assignments

1 Opdracht: Netflix en HBO

Open het model met de naam ‘Netflix en HBO’. Het model hoort overeen te komen met afbeelding 1.

1.1 Wat is de samenstelling van dit model? Omcirkel de juiste optie.

1. Het model bestaat uit 3 entiteiten, 3 proportionaliteiten en 4 hoeveelheden.
2. Het model bestaat uit 3 entiteiten, 3 proportionaliteiten en 2 hoeveelheden.
3. Het model bestaat uit 3 entiteiten, 2 proportionaliteiten en 3 hoeveelheden.
4. Het model bestaat uit 3 entiteiten, 2 proportionaliteiten en 4 hoeveelheden.

In de economie wordt er onderscheid gemaakt tussen complementaire goederen en substitutiegoo- deren.

**Complementaire goederen** worden vaak samen gekocht. Wanneer de vraag naar het ene product stijgt, stijgt de vraag naar het complementaire goed ook.

1.2 Voer een simulatie uit. Er verschijnen meldingen omdat het gegeven model niet klopt. Corrigeer het model door de afgeleide van HBO te veranderen (zodat hij niet meer afneemt), de 'Vraag' van Netflix moet op afnemen blijven staan.

1.3 Voer opnieuw een simulatie uit en streep het verkeerde antwoord door.

Als de vraag naar Netflix daalt, dan stijgt/daalt de vraag naar TV’s. Netflix en TV’s zijn comple-mentaire/substitutie gooderen.

_Sla het model op en ga door naar de volgende opdracht._

2 Opdracht: Zelfrijdende auto’s

_Open het model met de naam ‘Zelfrijdende Auto’s’. Het model hoort overeen te komen met afbeelding 2._

![Diagram van Zelfrijdende Auto's](image)

2.1 Wat is de samenstelling van dit model? Omcirkel de juiste optie.

*Tip: Gebruik de helpfuncties die zijn uitgelegd in de gebruiksaanwijzing.*

1. Het model bestaat uit 6 entiteiten, 3 proportionaliteiten en 5 hoeveelheden.
2. Het model bestaat uit 5 entiteiten, 3 proportionaliteiten en 4 hoeveelheden.
3. Het model bestaat uit 5 entiteiten, 4 proportionaliteiten en 3 hoeveelheden.
4. Het model bestaat uit 6 entiteiten, 4 proportionaliteiten en 3 hoeveelheden.
Het overgrote deel (90%) van verkeersongelukken komt door menselijke fouten. Hoe meer zelfrijdende auto’s deel zullen worden van het verkeer, hoe minder ongelukken er zullen zijn door rijden onder invloed, vermoeidheid of afleiding. Hierdoor zal het aantal fatale ongelukken in het verkeer afnemen. Deze positieve ontwikkeling heeft echter een kanttekening. Wanneer het aantal sterfgevallen in het verkeer afneemt, neemt namelijk ook het aantal orgaandonoren af. Sterfgevallen door verkeersongelukken zijn namelijk een van de meest voorkomende doodsoorzaken van orgaandonoren. Dit zal ertoe leiden dat er minder mensen hun broodnodige orgaantransplantatie ontvangen en hierdoor zal de kans op overlijden tijdens het staan op de wachtlijst voor een orgaandonatie toenemen.

2.2 Zet het aantal zelfrijdende auto’s op toenemen en voer een simulatie uit. Het simulatie effect komt niet overeen met de verwachting uit bovenstaande tekst. Achterhaal waar dit aan ligt door na het uitvoeren van de simulatie op de groene pijltjes te klikken. Corrigeer het model en noteer waar de fout aan lag.

Tip: De fout kan niet gecorrigeerd worden door de waardes van de afgeleides aan te passen.

Dankzij ontwikkelingen op technologisch gebied is de verwachting dat het in de toekomst mogelijk zal zijn om organen te printen met een 3D-printer. Deze uitvinding zal het aantal sterfgevallen van mensen op de wachtlijst voor een orgaandonatie doen afnemen.

Figuur 1: Een orgaanprinter
2.3 Voeg in DynaLearn de juiste proportionaliteit (zie figuur 2.3) toe tussen 'Aantal' van 'Orgaanprinters' en de hoeveelheid sterfgevallen op de wachtlijst voor orgaandonaties. Geef met een plus of min in de afbeelding hieronder aan welke proportionaliteit jij hebt toegevoegd.

2.4 Of het printen van organen de afname aan orgaandonaties door zelfrijdende auto’s kan compenseren hangt af van de effectiviteit en de kosten van orgaanprinters. Voer een simulatie uit met een toenemend aantal zelfrijdende auto’s en een toenemend aantal orgaanprinters. Welke situatie vind jij het meest voordehandliggend en waarom? Omcirkel in onderstaande afbeelding de uitkomst die met jouw verwachting overeenkomt en leg uit waarom jij deze uitkomst het meest waarschijnlijk vindt. Tip: Gebruik de hulpfuncties om te achterhalen waarom er drie verschillende uitkomsten zijn.

Sla het model op en ga door naar de volgende opdracht.
3 Opdracht: Leefbare zone

Open het model met de naam ‘Leven op Woestijnplaneten’. Het model hoort overeen te komen met afbeelding 3.

3.1 Geef aan of de volgende uitspraken over het model waar zijn:

1. Er zijn meer entiteiten dan hoeveelheden in dit model: Waar/Onwaar
2. Dit model bevat momenteel alleen maar positieve proportionaliteiten: Waar/Onwaar

Wetenschappers berekenen vaak de leefbare zone van een planeet om in te schatten hoe hoog de kans is dat er leven voorkomt op de planeet. De **leefbare zone** van een planeet is de afstand die een planeet kan hebben tot een ster waarbij het mogelijk is dat er leven zoals op Aarde voorkomt. De belangrijkste factor hierbij is de temperatuur van het water. Als het water niet bevriest of verdampt wordt ervanuit gegaan dat de planeet leefbaar is. De leefbare zone rondom de Aarde is in figuur 2 weergegeven.

In ons sterrenstelsel worden steeds nieuwe planeten ontdekt met een woestijnklimaat, zoals weergegeven in figuur 3. Aan jou de taak om te onderzoeken of deze planeet een grote leefbare zone heeft.
3.2 Denk jij dat het feit dat er weinig water op een woestijnplaneet aanwezig is ervoor zorgt dat de leefbare zone juist groot is of klein? Beargumenteer jouw vermoeden.
Uit jouw onderzoek naar de invloed van water op het broeikaseffect van een planeet zijn de volgende punten gebleken:

1. Wanneer er veel waterdamp in de atmosfeer van een planeet is, zal het broeikaseffect sterker zijn.
2. Als de hoeveelheid water toeneemt op een planeet, zal de hoeveelheid waterdamp in de atmosfeer ook toenemen.

3.3 Voeg twee proportionaliteiten toe die deze verbanden uitdrukken. Geef met een plus of min in onderstaande afbeelding aan welke proportionaliteiten je hebt toegevoegd, geef ook de richting van de proportionaliteiten aan.

3.4 Onderzoek welke hoeveelheid moet afnemen, stabiliseren of toenemen om de 'Grootte van de leefbare zone' te doen toenemen. Geef de naam van de hoeveelheid en de waarde waarop deze gezet moet worden. (Je hoeft alleen de waardes van de blauw pijl aan te geven en niet van de groene.) Zet ook daadwerkelijk de hoeveelheid naar de genoemde waarde, en controleer of bij het simuleren de 'Grootte van de leefbare zone' toeneemt.
3.5 Heeft jouw onderzoek in DynaLearn jouw vermoeden bevestigt of juist niet? Begrimeente de stelling ‘Als er meer water op een planeet is, is de leefbare zone groter’ aan de hand van het model in DynaLearn. Noem de vier hoeveelheden (Hoeeelheid water/Hoeeelheid waterdamp in de atmosfeer/Broeikaseffect/Grootte van de leefbare zone) in jouw antwoord.

Sla het model op en ga naar de link om de vragenlijst te beantwoorden.
Evaluatie

Ik ben benieuwd naar hoe jij de uitleg in DynaLearn hebt ervaren!

*Vereist

1. Vul hier jouw toegewezen e-mail adres in (dat begint met dynalearn.kkc) *

Vragen over jouw achtergrond

2. Mijn profiel is *
   Markeer slechts één ovaal.

   C&M  
   E&M  
   N&G  
   N&T

3. Ervaring met computers *
   Markeer slechts één ovaal per rij.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
   Ik gebruik de computer vaak om te leren.  |   |   |   |   |
   Ik creëer vaak diagrammen op de computer. |   |   |   |   |
   Ik vind het makkelijk om een programma als DynaLearn te gebruiken. |   |   |   |   |

Vragen over de uitleg

De uitleg in DynaLearn werd telkens weergegeven in witte spraakbubbel. Geef bij elke stelling aan in welke mate jij ermee eens bent. Als jij geen antwoord weet op de open vragen mag jij die leeg laten.

4. Weergave van de uitleg *
   Markeer slechts één ovaal per rij.

<table>
<thead>
<tr>
<th>Zeer mee eens</th>
<th>Oneens</th>
<th>Neutraal</th>
<th>Eens</th>
<th>Zeer mee eens</th>
</tr>
</thead>
</table>
   De uitleg is makkelijk in gebruik. |   |   |   |   |   |
   Het aanroepen van de uitleg is logisch. |   |   |   |   |   |
   De uitleg wordt logisch weergegeven. |   |   |   |   |   |
5. Indien van toepassing, wat zou jij veranderen aan de weergave van de uitleg?

6. Inhoud van de uitleg *
Markeer slechts één ovaal per rij.

<table>
<thead>
<tr>
<th>Ziekmee oneens</th>
<th>Oneens</th>
<th>Neutraal</th>
<th>Eens</th>
<th>Ziekmee eens</th>
</tr>
</thead>
<tbody>
<tr>
<td>De uitleg is goed te volgen.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>De uitleg is volledig.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>De uitleg bevat geen overbodige tekst.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Indien van toepassing, wat miste jij aan de uitleg?

8. Indien van toepassing, welk gedeelte van de tekst in de uitleg kan volgens jou weggelaten worden?
9. Effectiviteit van de uitleg *

Markeer slechts één ovaal per rij.

<table>
<thead>
<tr>
<th>Zonder de uitleg zouden de opgaven niet te maken zijn.</th>
<th>Zeer mee oneens</th>
<th>Oneens</th>
<th>Neutraal</th>
<th>Eens</th>
<th>Zeer mee eens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door de uitleg snap ik beter wat de elementen in DynaLearn voorstellen.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Door de uitleg snap ik beter hoe de elementen in DynaLearn elkaar beïnvloeden.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Door de uitleg kan ik het maken van fouten in het modelleren in DynaLearn beter voorkomen.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Door de uitleg snap ik beter wat het verschil is tussen een complementair en substitutie goed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Door de uitleg snap ik beter wat het effect van zelfrijdende auto’s is op orgaandonaties in de zorg.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Door de uitleg snap ik beter wat voor invloed water heeft op de leefbare zone van een planeet</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. Indien mogelijk, geef een voorbeeld van hoe de uitleg heeft bijgedragen aan het begrijpen van de elementen in DynaLearn en hoe de elementen elkaar beïnvloeden.

11. Rangschik de uitleg van meest naar minst nuttig *

Markeer slechts één ovaal per rij.

<table>
<thead>
<tr>
<th>Wat-uitleg: De definities van de elementen</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waarom-uitleg: De uitleg van de simulatieresultaten</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Waarom niet-uitleg: De uitleg die verscheen wanneer er een fout zat in het model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hoe-uitleg: De uitleg die vertelt hoe jij de waarde van een afgeleide kan veranderen na een rechtsklik op die waarde</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12. Indien jij nog algemene opmerkingen hebt over de uitleg of DynaLearn kun jij die hier invullen.